T.E.R : Economie du Logiciel Libre

LEONARD Landry

Mars 2004 – responsable Matthieu Neveu

Le Logiciel Libre, stratégie d'entreprise viable ?

« *Nous restons cependant dubitatifs sur le modèle économique du libre : jusqu´à quel stade une économie peut-elle supporter le bénévolat ? A un moment donné, qui sera bientôt atteint, on obtiendra un point d´équilibre entre logiciels libres et logiciels commerciaux. En attendant, oui, Linux mais également les autres logiciels libres sont des concurrents sérieux. Ils nous poussent à nous différencier par la valorisation fonctionnelle et économique de nos solutions*. » **Olivier Ezratty**, directeur de la division **Microsoft** .NET

Email : leonard.landry@free.fr

L'Université de Lyon II, filière économie quantitative, n'entend donner aucune approbation ni improbation aux opinions émises dans ce mémoire, ces opinions devront être considérées comme propres à son auteur.

Dans un monde où l'argent est omniprésent, rendre des biens « libres ou gratuits » peut s'avérer être une stratégie non pertinente car à priori non viable. Or dans nos économies à la recherche perpétuelle de profit, comment expliquer le quasi don de produits ou encore laisser libre d'accès une technologie. Il nous faut donc déterminer les raisons d'utilisation de tels procédés, comme dans l'industrie du logiciel, base de notre réflexion.

Peux-t-on vivre correctement du logiciel libre ?

Les avis divergent sur la question, Microsoft défend avec conviction son Commercial Software Business Model (logiciel propriétaire) face à l'Open Source Business Model (logiciel libre).

Au vu de la stratégie de l'entreprise IBM, l'open source peut avoir un attrait commercial indéniable (1 milliard de dollars d'investissement en 2001). Les résultats financiers du groupe sur les activités "logiciels" et "services" croissent depuis 3 ans (cf. annexe : *Résultat financier d'IBM*). En effet depuis cette date, le groupe a fait de Linux et des logiciels libres une composante essentielle de sa stratégie d'entreprise, il semblerait donc que celle-ci est été efficace pour la firme, en l'associant à une importante campagne marketing durant cette période. IBM n'est pas la seule entreprise à utiliser le concept d'Open Source.

Nous pouvons citer les firmes Red Hat (développeur linux), Apple (fabrication de matériel et de logiciel propriétaire), ou encore Sun (matériel et logiciel d'entreprise) ; ces trois firmes, elles aussi réorientant certes, de manières différentes, leur stratégie à l'aide du libre.

Ainsi, nous essaierons de comprendre et de déterminer les caractéristiques permettant aux entreprises de logiciels d'être viables tout en utilisant des solutions Open Source au sein de leur stratégie d'entreprise. Nous chercherons par là même le rapport entre industrie libre et propriétaire (complément ou substitut?).

Dans une première partie, nous ferons un état des lieux des théories économiques et de l'industrie Open Source. Ainsi, la première section nous permettra d'expliciter l'utilisation des licences, armes juridiques et stratégiques de protection, de développement de l'OSS et les choix de celles-ci. Puis, nous ferons un tour d'horizon des méthodes de génération et d'appropriation de revenus, gage de survie de la firme. Enfin, nous étudierons un modèle de contribution aux biens publics, afin de comprendre les intérêts de la contribution à un projet Open Source et la nécessité de coordination.

Dans un second temps, nous essaierons de comprendre en quoi l'OSS peut être une stratégie « intéressante », au travers d'exemples issus de l'industrie du libre d'accès. Les deux premières sections nous permettrons d'appréhender l'OSS comme armes accès et d'implantation sur le marché. Nous rechercherons ensuite à déterminer les intérêts qu'une firme peut retirer de l'utilisation de stratégie basée sur l'OSS, en tenant compte de la taille, de la structure de marché, ou encore de la législation en vigueur, afin de mieux comprendre les enjeux du libre.

Dans cette approche de l'économie de l'Open Source, les concepts d'incitation, de coopération et de coordination des agents au sein d'un groupe ne seront utilisés que pour servir à notre problématique. Nous en donnerons tout de même quelques notions, celles-ci aidant à la compréhension du logiciel libre en tant que fait économique.

Avant cela, nous devons déterminer les bases de l'OSS (Open Source Software) en donnant une définition précise de ce concept, souvent élargi à tort au « gratuiciel ». Par la suite, l'OSS n'étant connu que dans ses grands principes, il nous semble être essentiel pour une bonne compréhension du sujet de donner quelques éléments sur l'économie du logiciel libre.

Open ne signifie pas gratuit

En premier, il est trop souvent associé le terme gratuit lorsque l'on parle de logiciel libre. En effet l'OSS (Open Source Software) ou encore Free Software est souvent traduit par logiciel gratuit (or « *gratuiciel* » correspond au terme anglais *freeware*). Si le terme libre (free) peut porter à confusion, il nous faut donc reprendre la définition de l'OSS pour bien comprendre cette nuance. Ainsi ce n'est pas le logiciel en lui-même qui est libre mais son code source (code de programmation en langage humain compréhensible par le développeur, qui est ensuite compilé dans un code binaire traitable par l'ordinateur).

Ce code source peut donc être comparé à une recette de cuisine, ou encore à des plans de fabrication.

La notion de gratuité du libre vient de la facilité d'acquisition de ce code source, car ce dernier est ouvert d'où le terme Open Source. Les coûts d'accessibilité sont quasiment inexistants dans l'OSS à contrario des logiciels Closed Source (logiciel propriétaire) qui gardent secret ce code gageant un avantage technologique, et ne fournissent que l'exécutable.
En effet, pour pouvoir modifier un programme, le développeur/utilisateur doit avoir la possibilité de réécrire en partie ce code source, il doit ainsi désassembler l'exécutable (du langage machine au langage humain) afin de le retravailler. Or cette opération est très

complexe et peu donc s'avérer coûteuse si le logiciel est propriétaire. L'Open Source fournissant le code dans les deux langages, cette opération devient « gratuite » (or temps de programmation).

Nous voyons donc que le terme de gratuit ne doit être associé qu'au code source, et non au logiciel.

Open Source : diffusion et attrait

Le style bazar (Raymond, 1998) va à l'encontre de tous les usages convenus : il suggère de distribuer très rapidement un programme qui n'est pas nécessairement terminé, pouvant être grossier, bogué, imparfait ou mal documenté, et d'inciter ses utilisateurs à participer à son développement. Les utilisateurs ne sont plus alors que de simples consommateurs mais deviennent, chacun à leur mesure, co-développeurs du programme, repérant les erreurs, les corrigeant parfois ou y ajoutant de nouvelles fonctionnalités....

La communauté du libre justifie la présence et l'utilisation de logiciels libres car ils disposent de qualités supplémentaires face au logiciel propriétaire (qui ne peuvent pas rassembler les six) :

- la qualité
- la réactivité
- la pérennité
- le coût
- la liberté
- la concurrence

Open Source ne signifie pas Linux:

Si le logiciel Linux (OS : operating system) est devenu le logiciel type de la communauté grâce à son évolution tant dans sa diffusion que dans ses contributions, le libre ne s'arrête pas à ce système d'exploitation alternatif.

Nous pouvons citer en exemple d'autres réussites du libre parmi les 12000 projets actifs (Ghost, Prakash, 2001) : FreeBSD (OS), Apache (web serveur), SendMail (routage du courrier électroniques), Certaines versions d'Unix (OS), Langages script PERL et Ruby (programmation), T_EX (traitement du texte)….

I/ L'OPEN SOURCE : du juridique à l'économique
Etat des lieux et théories économiques

A : Les licences : armes stratégiques de l'Open Source

Les licences s'assimilant à des contrats d'utilisation du logiciel, elles définissent les droits et devoirs de l'utilisateur, ainsi que ceux du développeur ; ainsi leur spécificité peut amener à des résultats économiques différents. Le choix de la licence peut donc être stratégique, l'étude de ce dernier nous semble donc nécessaire.

Si l'utilisateur ne lit que brièvement la licence lors d'une installation d'un logiciel propriétaire ou Open Source, son rôle est pour nous essentiel. En effet, elle pose le cadre juridique qui régit la vie du logiciel (utilisation, modification,…).

Or, dans le cas de logiciel propriétaire, la firme se réfère aux pratiques de la concurrence et intègre ses attentes à ce contrat la liant avec l'utilisateur mais aussi avec la concurrence.

Au début des années 80, Richard Stallman, chercheur au M.I.T. et fondateur de la Free Software Foundation, crée la base des licences de type copyleft*. Ainsi après les déboires du développement de l'OS UNIX, lancé sous un system Open Source, puis revendiqué par son créateur (AT&T); il chercha à définir un cadre de développement d'un système d'exploitation complet et libre (GNU's not UNIX). Dans ce cadre, il rédigea la GNU-GPL (General Public Licence). Cette construction juridique se base sur le copyright afin de protéger l'auteur mais intègre des restrictions pour préserver la liberté de réutilisation, de modification et de reproduction. Ainsi un logiciel libre doit obligatoirement être fourni avec son code source et l'ensemble de la documentation permettant à chacun de le comprendre et de l'améliorer.

Selon les sources, on estime que 80% des logiciels Open Source sont développés sous licence GPL (cf. annexe *Licences*). Ce chiffre s'explique principalement pour deux raisons ;
En premier lieu cette licence a pour avantage d'être la première à avoir été développée pour l'Open Source et respecte au plus près la volonté de la communauté, c'est-à-dire la liberté du logiciel comme explicitée précédemment.

---

* La Free Software Foundation considère qu'une licence est libre (copyleft) si elle permet:
- d'exécuter le programme librement pour tous les usages
- d'étudier le fonctionnement et de l'adapter le programme à ses besoins
- de redistribuer des copies
- d'améliorer le programme et de publier ses améliorations

Second point important, cette licence peut être qualifiée de virale. En effet celle-ci impose que tout logiciel se basant ou intégrant tout ou partie d'un code source sous licence GPL doit être développé sous licence GPL. Cette caractéristique de contamination peut donc expliquer la forte utilisation de ce type de licence.

Bien que très utilisée la licence GPL a pour principal défaut ses caractères viral et trop « libre ». En effet, un logiciel libre sous GPL ne permet pas de créer un avantage produit basé sur une technologie (méthode de programmation), puisque les sources sont disponibles, re-distribuables et modifiables à souhait, y compris par un concurrent. Dès lors, tirer un revenu de la vente de logiciel Open Source est très difficile et une activité complémentaire de service (maintenance, débogage…), reste très aléatoire sur un marché concurrentiel (marges faibles). Nous démontrerons ce point plus loin avec l'exemple de la nouvelle stratégie de Red Hat. Malgré ses défauts, la licence GPL a pour principal atout de permettre une diffusion maximale du code et donc d'augmenter l'implantation et les contributions.

Il existe aussi une variante du GPL, le LGPL (Lesser General Public Licence) qui est, comme son nom l'indique, une version moins contraignante. Elle permet d'utiliser le programme, tout du moins en partie, en le mixant avec d'autres logiciels ou routines sans que ceux-ci ne soient forcément contraint de devenir Open Source (Lerner et Tirole, 2002). Cette licence a été écrite principalement pour des bibliothèques de programmation, afin de permettre leurs intégrations à des logiciels dont la licence n'est pas compatible avec la GPL (logiciel propriétaire).

Si la GPL reste la licence la plus utilisée, plusieurs alternatives ont été introduites par les développeurs. Ainsi, nous devons garder à l'esprit que le choix de la licence incombe au(x) créateur(s) développeur(s) du projet Open Source, elle peut donc dépendre des attentes de ce(s) dernier(s).

L'alternative principale au GPL est la licence BSD (Berkley Software Distribution). Cette licence copyleft permet une appropriation (non diffusion) des modifications par leur(s) auteur(s).

Ex : BSDfree, Apache (sous une licence dérivé du BSD…)

Lorsque le projet Open Source est à l'initiative d'une firme commerciale, c'est-à-dire en général l'ouverture d'une partie du code source d'un logiciel propriétaire, celle-ci crée une licence spécifique (Apple Public Source Licence : développer lors de l'ouverture du code source d'une partie de Mac OS X par Apple, ou encore la Sun Community Source Licence pour son logiciel Java, ….).

Après avoir défini les principales licences du monde de l'OSS, nous devons nous questionner sur les répercutions du choix de celles-ci.

Ainsi, une licence peut avoir une influence sur le comportement de la communauté des programmeurs du libre. Plus la licence est proche de la définition stricte d'OSS, plus ces derniers sont encouragés à participer à un projet, celui-ci ne pouvant pas être détourné de ses objectifs principaux et donc exploité par des entreprises profitant de la main d'œuvre gratuite de la communauté.

Si le choix de la licence a des impacts sur les développeurs, elle influence aussi les utilisateurs par une diffusion indirecte. En effet, la qualité du logiciel variant avec les contributions des développeurs, si la licence n'incite pas les programmeurs à contribuer au projet, l'utilité retirée par l'utilisateur d'un logiciel non terminé sera diminuée (donc sous optimale). De plus la licence peut influer sur la durée de vie du projet ; une sur commercialisation ou une difficulté d'implantation (logiciel GPL ne pouvant être intégré dans des logiciels propriétaires du au caractère virale de la licence) peuvent entraîner l'arrêt du développement du logiciel. Inversement, une commercialisation restreinte par la licence peut ne pas être efficiente pour le développement du logiciel ; certains utilisateurs, étant prêts à acheter logiciel et service découlant de projet Open Source, doivent pouvoir trouver des entreprises répondant à cette demande.

Nous voyons donc que le choix de la licence a des répercutions sur les motivations des développeurs, des utilisateurs et des firmes. Elle constitue un outil stratégique incontournable du logiciel libre. Reste que si le « Licensor » (le développeur seul ou en groupe responsable du choix de la licence [Lerner et Tirole, 2002]) détermine la protection de son innovation en fonction de ses préférences, il ne peut tout de même pas faire abstraction de l'avis de la communauté des utilisateurs et des programmeurs.

B : La génération de revenus, gage de survie de l'Open Source

Dans cette partie, nous allons essayer d'analyser les possibilités offertes à la firme de création de flux futurs garantissant sa pérennité, dans le cadre d'une stratégie Open Source. Si la licence donne le cadre d'utilisation et de production du logiciel, elle n'assure pas entièrement les revenus.

Selon Linus Dahlander, les firmes disposent de cinq méthodes de génération de revenus :

- La consultation: expertise et conseil
- L'éducation: formation des utilisateurs
- Le support: aide technique (résolution de conflit, assistance, maintenance)
- La licence : ajouter des parties propriétaires à un code source libre
- La Black-Box: lier l'OSS à une solution de matériel

Ces cinq méthodes peuvent être assemblées au gré de l'entrepreneur afin de maximiser ses sources de revenus.

Si l'auteur donne l'exemple de six firmes (Start-up Nordiques), nous pouvons très facilement trouver des stratégies de grandes entreprises utilisant une ou plusieurs de ces méthodes.

Red Hat package ses solutions linux en incluant un support (tout public), une documentation (manuel ou cd « éducatif ») et, pour les entreprises, fournir un service d'expertise (analyse des besoins, des structures informatiques). En effet, sur le marché des distributions linux la concurrence est importante, ce qui ne permet pas de vendre le logiciel seul avec une marge confortable (cf. : *part de marché des distributions linux* en annexe). Ainsi, les revenus sont assurés par la fourniture de documentation, de formation, de support technique, mais aussi d'analyse des besoins des firmes. Nous voyons donc que Red Hat utilise les trois premières méthodes citées, les autres étant pour lui peut accessibles car nécessitant de fournir du matériel, ou de modifier la licence du logiciel (linux étant sous GPL cela est impossible).

Pour voir apparaître l'utilisation des deux dernières méthodes citées, il nous faut nous intéresser à d'autres firmes susceptibles de pouvoir les utiliser.

L'entreprise IBM (constructeur informatique) utilise la méthode de Black-Box, elle groupe donc matériel à OSS (et par là même software à support, éducation pour les mêmes raisons que la firme Red Hat). En effet, ce constructeur surfe sur la vague du libre en associant son matériel à des logiciels open source, il réduit les coûts de ses solutions tout en s'assurant de nouveaux revenus par adjonction des nouveaux services.

Enfin, Apple utilise en plus du Black-Box, la méthode licence. Ainsi le constructeur et développeur des MACs, a récemment mixé le closed et l'open, afin de se garantir des revenus sur le logiciel Mac OS X (cf. : partie 2C).

Nous voyons donc au travers de ces trois exemples, que la survie du libre peut être garantie par les activités annexes de services ou de ventes de produits complémentaires (matériel, logiciels utilitaires…).

Une autre source de revenu peut être le logiciel en lui-même, en protégeant au plus le logiciel mais en le gardant l'esprit « libre ».

Méthodes de protection de l'OSS suivantes (Dahlander):

- Le secret : ajouter des parties propriétaires à un code source libre (comme les licences au dessus)
- Le first mover : créer l'avantage d'une présence historique (s'implanter avant les autres)
- Les externalités de réseau : utiliser la communauté pour influencer l'utilisateur anticipant les contributions de celle-ci à conserver la distribution de la firme (voire à l'acquérir, cela devant une source de revenu)

Les trois méthodes décrites ci-dessus permettent à l'entreprise de se créer un avantage produit (source de rente) et les méthodes de génération de revenu permettent de produire des plus produits (générateurs de revenus annexes).

C : Le logiciel libre et les difficultés de production d'un bien public

Les sections précédentes nous ont permis d'appréhender la protection du libre puis l'acquisition de revenus, or pour que ces étapes soient nécessaires le logiciel doit avoir été développé (produit). Nous devons donc étudier la méthode de production du libre qualifié par certains auteurs de bazar.

En effet, le mode de production du libre décrit par Raymond (1998), met en avant le rôle de l'incitation et de la coordination, dans la contribution à un logiciel libre.

Un bien public est définit dans la théorie économique selon deux critères. Il doit en premier être non rival, c'est-à-dire que sa consommation par un agent ne réduit pas celle des autres, mais doit aussi être non exclusif, le bien pouvant être consommé ou utilisé par tout le monde sans exclusion. Ainsi un bien respectant ces deux conditions est considéré comme un bien public pur.

L'OSS peut-il être considéré comme un bien public ?

Le caractère non rival du logiciel découle de sa production, en effet les faibles coûts de duplication permettent très facilement de satisfaire la demande. Ce caractère s'est encore renforcé avec le développement d'Internet, le réseau mondial permettant un échange du producteur au consommateur sans même de support physique. Cette notion n'est pas valable que pour les logiciels Open Source, car la production ne diffère pas de l'industrie globale du software (Microsoft peut dupliquer et donc fournir ses produits aussi aisément qu'une firme Open Source).

En revanche, le caractère non exclusif ne peut s'appliquer qu'à l'Open Source. Le code source du logiciel étant ouvert, son appropriation est simplifiée et n'entraîne aucune exclusion (la licence servant de garde-fou et maintenant ce principe de liberté d'accès).

Au vu de cela, le logiciel libre peut se voir qualifié de bien public, ce qui nous contraint à nous pencher sur les méthodes de production et donc sur les notions d'incitation, de coopération et de coordination.

Au regard de la théorie des jeux et des méthodes décentralisées de développement (communauté Internet), nous pouvons nous demander comment ce bien public particulier qu'est le libre peut être produit efficacement. En effet, avec la présence d'externalité positive sur un marché, l'utilité des individus augmente avec l'utilisation par d'autres d'un bien (substitut ou complément, voire même sans lien direct). Nous pouvons prendre l'exemple d'un accès Internet : plus il y a d'utilisateurs plus le nombre de sites et les possibilités de communication (mails, chats, forums…) se développent, augmentant par là même son attrait.

Ce phénomène est définit par la théorie comme une externalité de réseau. Or, une telle externalité se retrouve dans l'OSS, les possibilités du logiciel augmentent avec le nombre de développeurs (effet contribution) et d'utilisateurs (effet implantation). L'effet contribution crée une externalité car les co-développeurs s'échangent des tâches mais aussi un savoir-faire (technique de programmation), ainsi le travail fait par l'un bénéficie à tous. Les utilisateurs profitent eux aussi de ces améliorations et en adoptant le logiciel contribuent à faire grandir le projet.

Les effets implantation et contribution bénéficient donc d'externalité positive et réciproque, car l'incitation d'un programmeur (faire reconnaître son travail) augmente avec la taille du projet celui-ci dépendant de l'utilisation, elle-même finalité des contributions.

En se basant sur les papiers de Lerner et Tirole, les incitations découlent des points suivants :

- Amour de la programmation (acte de loisir)
- Besoin de résultats (intérêt personnel à la tâche)
- Désir de contribuer (envie de partager ses connaissances dans la programmation)
- Participation à une communauté (sentiment d'appartenance)
- Envoi d'un signal (expression de ses compétences au moyen du libre)

Ces incitations non monétaires semblent être acquises par les chercheurs de l'économie du libre (consensus de la littérature sur leur définition), et peuvent expliquer en partie la collaboration des développeurs à un projet. En effet, chaque développeur individuel du libre (sans intérêts financiers) participant à un projet Open Source peut reconnaître dans les cinq propositions précédentes celle(s) qui lui correspond(ent).

En revanche, elles n'expliquent les participations qu'à titre particulier, mais ne décrivent en rien celles des firmes. Les propositions faites peuvent être des motivations lors de la création par l'entrepreneur (souhait de produit de qualité, créer une communauté alternative…), mais cela ne nous dit pas pourquoi des stratégies basées sur l'OSS sont lancées ou poursuivies.

A cela, nous pouvons ajouter la question de la coordination, qui ne se règle pas par les incitations. En prenant l'exemple du signal, tous les agents, souhaitant démontrer leur capacité, vont contribuer aux tâches complexes et délaisser les modules « non-sexy » où les compétences sont moins valorisées (partie user friendly, et interface graphique…), le projet fait donc face à une incomplétude des contributions par défaut de coordination.

Le modèle mis en évidence par Myatt et Wallace (2002), dans « *Equilibrium selection and public good provision : the development of open source software* », nous permet très simplement de comprendre les limites de coopération à l'OSS par les individus.

Pour cela, les auteurs ont modélisé un jeu de contribution au bien public (OSS), représentant deux agents (**i ϵ {1,2}**) faisant face à un choix binaire : contribuer (C) ou non (D). Cette restriction à un choix binaire est volontaire car elle évite l'apparition d'un nombre de combinaisons trop important à inclure dans le raisonnement.

Si l'individu contribue, il lui en coûte c_i qui n'est pas influencé par le comportement de l'autre. On suppose aussi que les coûts sont différents pour les deux agents ($c_1 \neq c_2$).

Dans le cas ou un seul bien public pourrait être produit, nous voyons apparaître le problème de Free Riding, un individu ayant intérêt à laisser contribuer l'autre.

Bien public unique (nécessitant au moins 1 développeur)

	C	D
C	$v-c_2$ $v-c_1$	v $v-c_1$
D	$v-c_2$ v	0 0

v : gains attendus par l'individu (monétaire ou non)

Il apparaît ainsi que si **$v<\min(c_1, c_2)$**, les gains des 2 joueurs ne sont pas compensés par les coûts en cas de contribution, l'équilibre est donc {D, D} et les joueurs n'interviennent pas dans la production du bien public. Le coût supporté par les deux individus est supérieur au gain attendu, ainsi aucun n'a d'intérêt à contribuer.

Dans le cas où **$v>\max(c_1, c_2)$**, les deux joueurs ont un profit quand le bien public est créé, qu'ils contribuent ou non (si 1 contribue 2 gagne qu'il joue C ou D, il en est de même dans l'autre sens). Nous voyons ici que collectivement le bien doit être produit, mais individuellement chaque agent a intérêt à ne pas contribuer car il fait un gain sans le moindre coût. Il apparaît dans ce jeu, deux équilibres en stratégie pure {C, D} et {D, C}, nous voyons ainsi se dévoiler la nécessité d'une coordination. Dans ce premier jeu où seul 1 contributeur suffit, une question ressort : qui contribue au bien public ?

Pour recoller à notre sujet, nous voyons en cela la difficulté des firmes à contribuer dans un projet OSS, car leur profit s'améliorerait en récupérant, quasiment sans coût, le bien public créé (code source) par une autre. Si le Free Riding semble être bénéfique, la firme qui ne contribue pas directement, perd l'avantage du premier entrant ou « first mover » car elle attend les résultats des efforts de l'autre. La firme qui contribue seule, bénéficie des résultats (code source donc logiciel) en premier, et peut donc débuter son implantation sur le marché avant les outsiders. Or ceci est un avantage très important face à la concurrence, car les consommateurs (entreprise ou particulier) ont un coût au changement (transfert de données, mise en attentes des requêtes aux serveurs…), ainsi le premier en place obtient une probabilité supérieure de fournir ultérieurement de nouvelles versions ou de nouveaux services.

Nous supposons dans ce jeu que pour obtenir le bien public **pur,** il n'y a pas de restriction (pas d'exclusion). Or, comme nous avons vu dans la première section, les licences peuvent restreindre les appropriations et donc conduire des coûts supplémentaires pour le free rider qui ne sont pas pris en compte dans ce jeu.

2 Biens publics (un développeur par bien)

	C	D
C	$2v-c_2$ $2v-c_1$	v $v-c_1$
D	$v-c_2$ v	0 0

Dans cette version du jeu, les auteurs décrivent les deux biens à produire comme indépendants (non substituables et non complémentaires), cela leur permet de dire que si un bien est produit, le gain de chaque individu est v auquel leur coût respectif doit être déduit (si contribution), et devient 2v si les deux biens sont produits.

Si **$v<\min(c_1, c_2)$**, l'équilibre se situe en {D, D}, même si quand $2v>\max(c_1, c_2)$, l'optimal serait {C, C}. Comme pour le premier jeu, le coût supporté par les deux agents est supérieur au gain attendu, ils n'ont donc aucun intérêt à contribuer.

Dans le cas où **$v>\max(c_1, c_2)$**, l'équilibre se situe en {C, C}, ainsi les contributions amènent à la production des deux biens. Les gains sont ici supérieurs aux coûts, il est donc optimal pour les agents de produire les deux biens, ainsi chacun doit contribuer à un bien afin que les deux soient créés et que le gain soit de 2v (v si les deux agents contribuent au même bien).

Nous voyons au travers de ce second jeu, apparaître le concept de coordination. Les deux individus doivent contribuer mais chacun à un bien différent, or le jeu étant instantané il n'y a pas d'apprentissage à travers les actions passées de l'autre, et rien ne permet d'anticiper à priori la contribution de l'autre à la production d'un bien.

Nous pouvons affirmer grâce à ce jeu qu'une bonne contribution, c'est-à-dire une participation à toutes les taches, n'est pas assurée et que le développement d'un projet peut échouer par manque de coordination. Nous voyons donc ici l'une des limites de l'OSS comme stratégie de firme. En effet, si la société attend les contributions de la communauté celle-ci n'est pas assurée pour toutes les taches au regard de ce jeu.

Mais ce jeu omet plusieurs aspects présents dans l'industrie du logiciel Open Source. Dans les faits, les acteurs du libre sont nombreux réduisant ainsi la probabilité qu'une tache ne trouve aucun contributeur. Les utilisateurs développeurs possèdent dans les faits des caractéristiques individuelles propres à chacun, or leur compétence et leur attrait pour la programmation varie avec les domaines d'application. Chaque agent contribue en fonction de ses préférences et de ses compétences, ainsi les fonctions de base (noyau de l'OS…) seront réalisées par les individus expérimentés et les modules « non-sexy » par des novices. De plus, la communauté est « en ligne », l'Internet permet de nos jours une coordination plus importante si non parfaite ; les sites, listes de diffusion, forums, chats de l'open source permettent à la communauté de se répartir les tâches, et de connaître les avancées de chacun afin de ne pas avoir de doublons…Ces échanges permettent une certaine coordination d'une communauté du libre qui ne peut donc réellement être décrite ni par un modèle cathédrale (planification centralisée), ni par un modèle bazar pur (Raymond, 1999).

Nous pouvons aussi voir que le modèle de jeu one shot de Myatt et Wallace n'est pas conforme à la réalité des projets Open Source. La programmation s'effectue en étape, (noyau puis modules annexes) ainsi le modèle devrait être élaboré autour d'un jeu séquentiel. Dans un tel cas, il reste à déterminer les stratégies de gâchette éventuelles permettant d'encourager la coopération (la présence de modérateur au sein de la communauté régulant l'échange).

Nous pouvons donc affirmer que la question de la coordination doit être un des points de réflexion d'une « bonne » décision stratégique (rentable), car elle assure la viabilité d'un projet Open Source.

Nous voyons que la stratégie de l'entreprise doit répondre au problème de coopération, d'incitation et de coordination, laissant par la suite à la firme le choix de la licence (nouveau produit logiciel) ou des méthodes de génération de revenus afin d'interagir sur le marché en se donnant ou non les moyens de vivre. La stratégie de l'entreprise est donc une donnée essentielle, car elle influence la survie du logiciel et donc de la firme. Le choix de la licence doit être efficient :

- protection trop forte : course à l'innovation et gaspillage de ressources
- protection trop faible : pas assez de ressources consacrées à l'innovation.

On voit que le choix de la licence pose les mêmes problèmes dans le libre que pour des logiciels propriétaires (question récurrente de la brevetabilité du logiciel).

Il en est de même pour l'appropriation de revenu qui doit être mûrement réfléchie, face au problème de free riding décrit précédemment. Et les entrants potentiels peuvent profiter du travail de l'insiders, entrer sans coût sur le marché et par la même s'octroyer des revenus anticipés par cette dernière.

II/ L'OPEN SOURCE : de la théorie à la pratique
Les applications stratégiques

A l'aide des sections précédentes, nous allons tenter de comprendre les stratégies « libre » envisageables par une firme. Trois points nous semblent importants à aborder. Nous étudierons successivement l'usage de logiciel libre pour l'accès au marché puis les stratégies d'implantation sur un marché qui en découle. Enfin nous nous interrogerons sur la viabilité de ces décisions stratégiques (à moyen et long terme).

A : OSS : arme stratégique d'accès au marché logiciel

Dans une économie, à marché donné avec un nombre d'acteurs déterminé, tout nouvel entrant rencontre une barrière à l'accès. Les entrants ne possèdent pas les avantages d'une technologie mature, de l'expérience ou d'une base de clients, c'est en cela que l'on peut définir la notion de barrière d'entrée. En effet, pour acquérir les mêmes avantages, le temps mais aussi le coût risquent d'être dissuasifs pour les entrants potentiels. C'est la difficulté à obtenir la même technologie, la même expérience ou les mêmes clients qui détermine donc l'existence ou non de barrières à l'entrée.

Ce que nous devons garder à l'esprit, c'est que les barrières à l'entrée sur un marché sont indispensables pour les insiders, car nécessaires au succès d'une entreprise. S'il n'y a pas de barrière à l'entrée suffisamment élevée, les entreprises n'arrivent pas à protéger leurs acquis et ne rentabilisent donc pas leurs investissements. Comme nous avons vu précédemment la firme choisit une méthode de génération de revenus, or si l'entrée de nouveaux concurrents est aisée, celle-ci entraînera une segmentation du marché et donc une diminution des profits individuels (baisse des prix des logiciels et des services annexes, baisse des quantités…).

A contrario, les barrières à l'entrée semblent aussi être des entraves à la concurrence, les outsiders ne pouvant pas librement (à coût raisonnable) accéder au marché. Ce manque de concurrence peut entraîner une stagnation de la qualité des produits, la course à l'innovation et donc les évolutions étant réduites car non nécessaires (les partisans de Linux reprochent à Microsoft de ne pas avoir fait évoluer son OS Windows comme la technologie le permettait, grâce à sa domination du marché). Il n'y a pas que la qualité qui peut être affectée mais aussi les prix des produits et des dérivés (services, supports….).

La présence ou non de barrière à l'entrée peut donc avoir d'importantes répercussions sur le marché. Si elles sont trop importantes, le manque de concurrence amène à une situation de monopole ou d'oligopole (risque de collusion) et donc à certaines inefficacités. Et si elles sont trop faibles, la viabilité du segment d'activité n'est pas assurée. Nous pouvons affirmer que les barrières à l'entrée sont un élément essentiel de la structure de marché à analyser pour une entreprise.

Après avoir brièvement défini ce concept de barrières d'entrée, il faut nous pencher sur la relation entre le logiciel Open Source et ce dernier. L'OSS permettant de lever certaines des barrières à l'entrée, il crée deux effets opposés. En premier lieu, il permet un accroissement de la concurrence bénéfique pour les consommateurs, mais il peut aussi engendrer de l'instabilité car les profits espérés peuvent se dégrader rapidement avec le nombre d'entrants.

En effet, le logiciel libre en ouvrant le code source et en le laissant libre d'accès, permet aux outsiders, mais aussi à des firmes déjà présentes sur le marché, de s'approprier une technologie quasiment sans coût. En cela, le développement d'un logiciel sous licence Open Source peut avoir un effet bénéfique pour l'utilisateur par une augmentation de la concurrence mais a pour effet d'atténuer les possibilités d'appropriation de revenus du développeur ou de la firme à l'initiative du projet. En revanche, il n'y a pas un effet négatif pour la partie développement de l'OSS, car une hausse de la concurrence peut permettre un développement du marché grâce à une augmentation de la qualité (contribution de nouveaux développeurs ou de nouvelles firmes) et de la notoriété du logiciel (hausse de la « publicité » par les sites, les forum des entrepreneurs).

Jusqu'ici, nous avons basé notre raisonnement principalement du côté des insiders, il nous faut donc explorer les effets sur l'accès au marché pour les entrants potentiels.

Ainsi, pour l'outsider, se baser sur une stratégie Open Source a pour effet positif de diminuer ses coûts de création, car après avoir développé la base du logiciel (noyau et modules annexes), la firme laisse à la communauté les patries amélioration et débogage du logiciel, qui sont les opérations les plus coûteuses du développement d'un logiciel (82% du coût de production [Cusumano, 1991]). De plus, grâce à Internet, le temps et les coûts d'échanges d'informations requis par ses opérations sont considérablement réduits. La firme n'a donc plus besoin de trouver nécessairement de distributeurs locaux de son offre logiciel, elle fournit elle-même ses produits grâce un site web qui sert aussi de moyen de communication et de promotion de ses technologies.

Au regard de cela, l'OSS peut être considéré comme un moyen de faciliter l'accès au marché. Mais si la levée des barrières d'entrée est effective pour l'entreprises se lançant dans un projet libre, il en est de même pour les autres outsiders qui eux peuvent bénéficier d'un code « prêt à l'emploi » sans même le coût de création du noyau originel, cette concurrence potentielle peut rendre le projet mort-né.

Ainsi, si le recours au logiciel libre dans une stratégie d'entreprise (par nécessité ou par choix) peut aider l'entrant potentiel à accéder plus aisément sur le marché. Ce dernier ne doit pas oublier que l'avantage ainsi obtenu n'est pas exclusif et peut bénéficier à d'autres, qui viendront diminuer ses possibilités de profits. C'est à cette étape stratégique qu'intervient le rôle du choix de la licence qui peut avoir des effets restrictifs nuancés, permettant par là-même de conserver un avantage produit permettant l'accès à une rente.

B : Les stratégies d'implantation

Nous avons vu dans la section précédente, que l'OSS permettait de lever les barrières à l'entrée des outsiders, en éliminant une partie des coûts dus à une quasi-externalisation d'un certain nombre de tâches de développement.

A cet accès simplifié au marché, il faut ajouter la possibilité d'une notoriété rapide découlant des contributions de la communauté du libre. En effet, un accroissement très rapide de la qualité du produit (par une augmentation du nombre de développeurs contribuant au projet) peut entraîner, dans le cas d'une réponse à un besoin, une standardisation par une adoption rapide de nombreux utilisateurs. Dans le cas où le logiciel s'attaque à un nouveau segment ou un nouveau besoin (portabilité, démocratisation, modularité, ou convivialité), une telle standardisation confère à la firme s'étant lancée dans le projet, l'avantage de first mover. Elle acquiert une réputation sur le marché en question (avantage d'une présence historique), et peut même disposer d'avantages technologiques si des parties du logiciel sont sous licence propriétaire.

Si le logiciel n'est pas nouveau dans son concept ou n'apporte pas de fonctionnalités nouvelles, il peut tout de même supplanter ses concurrents grâce à une « vision élargie du développement » ; chacun des contributeurs appartenant à la communauté n'étant pas supervisés, ils peuvent intégrer de nouvelles possibilités au logiciel qui n'étaient pas prévues à l'origine du projet (l'intégration d'un lecteur multimédia dans un OS linux peut être un exemple) ou encore au travers d'une logique de programmation différente, amener des

améliorations substantielles. Même avec une qualité inférieure, il peut y avoir implantation si le logiciel apporte un plus au marché. Un projet intéressant, même non fini, peut trouver sa place puis gagner des parts de marché (au sens du nombre d'utilisateurs).

Nous pouvons prendre l'exemple de Linux, dont la version initiale a été écrite en 1991 par un jeune informaticien finlandais Linus Torvalds. Ce logiciel, symbole du libre, n'était qu'un système d'exploitation basique* face au géant Microsoft Windows© (90% des ordinateurs équipés en standard) et est devenu en quelques années le concurrent avec le plus fort potentiel de l'OS le plus connu au monde. Cela est dû à l'appropriation par la communauté du libre de ce système se voulant alternatif aux solutions propriétaires (rappelons que linux est sous licence GPL, la plus représentative de « l'esprit libre »).

Nous pouvons ajouter à ce qui précède, l'effet prix des solutions Open Source. Pour implanter un nouveau produit, à qualité égale, il faut nécessairement que le coût de cette solution soit plus faible pour l'utilisateur. Or le coût de développement étant plus faible, le prix d'une solution Open Source est considérablement réduit. Ainsi les possibilités de diffusion semblent plus importantes, car l'acquisition du logiciel se révèle moins onéreuse pour l'utilisateur facilitant par là même l'adoption programme par le marché.

Attention, rappelons que coût réduit ne signifie pas que le logiciel est gratuit, mais que le prix est inférieur à une équivalence propriétaire** (qualité et fonctionnalité identiques [voir annexe]), les coûts de programmation étant extrêmement réduits dans un système Open Source.

***** Linux : un OS incomplet à son lancement :

"Linux is NOT portable (uses 386 task switching etc.), and it probably never will support any thing other than AT-hard disk, as that's all I have."

"Linux n'est PAS portable (utilise le changement de tâche du 386 etc.), et il ne supportera probablement jamais autre chose que des disques AT, étant donné que c'est tout ce dont je dispose." **Linus Torvalds**, 25 Août 1991

Actuellement Linux est compatible avec une majorité des périphériques de marque, et des types de systèmes (téléphone mobile, serveur, poste de travail….)

** Les offres de postes clients Linux de Mandrake, Red Hat, Novell/SuSE et Sun coûtent pour la plupart moins de 100 euros, de quoi motiver les entreprises face au tandem Microsoft Office 2003 PME édition et Windows XP Pro facturé 940 euros au détail.
(www.indexel.net)

Pour exemple, Open Office (suite bureautique) qui, avec un prix réduit et des fonctionnalités similaires, fera perdre environ 10% de part de marché à Microsoft Office en 2004 (prévision de Gartner).

Ainsi de telles solutions permettent aux intégrateurs, constructeurs ou détaillants de fournir sans frais des suites logicielles complètes à l'utilisateur. L'implantation de la firme dans l'ordinateur se fait ainsi sans même que le client en soit conscient. Or le logiciel est un produit périssable dans le sens où l'évolution de la technologie oblige une mise à jour. Après cette implantation sans frais, la firme bénéficie donc de clients pour d'éventuelles ventes de produits annexes.

Après avoir étudier les possibilités d'implantation offertes par une stratégie basée sur le libre qui sont nécessaires pour une frime, nous devons nous interroger sur la viabilité d'une stratégie Open Source qui est elle essentielle.

C : Les stratégies de MT LT : la viabilité de l'OSS

La question de la viabilité d'une stratégie s'impose forcément au(x) décideur(s), et l'Open Source ne faillit pas à cette règle. Or, l'utilisation du libre dans ses activités pose le problème des revenus.

Si une licence propriétaire garantit l'exclusivité de la distribution d'un logiciel par son développeur et donc par la même occasion des revenus potentiels, l'OSS ne les assure en rien. Ainsi l'Open Source limitant les barrières à l'entrée, la concurrence peut être vive ; surtout pour des logiciels basés sur les licences GPL (très libres) qui ne garantissent pas d'avantages technologiques à une firme (par appropriation de ses modifications).

La viabilité du libre dépend donc en partie du choix de la licence, une firme basant sa stratégie sur l'Open Source pour le lancement d'un nouveau produit doit donc retenir un certain degré de protection (licence type BSD,…), afin de se garantir un marché peu concurrentiel.

Pour une firme dont le logiciel libre est une activité complémentaire, elle peut obtenir des revenus des produits annexes (matériel, entretien, services…). Nous pouvons reconnaître ici la stratégie de la firme IBM qui, historiquement, fournissait son matériel avec ses propres Systèmes d'exploitation, et depuis quelques années a créé une gamme complète destinée aux entreprises (serveur, poste de travail…) sous Linux. IBM bénéficie ainsi de l'attrait de LINUX et de l'Open Source pour favoriser la vente de ses produits hardware et tire les

revenus confortables (voir annexe) de l'augmentation des ventes et des services associés (maintenance, mise à jour….).

« Nous ne sommes pas des profiteurs. Nous exploitons le travail de la communauté, mais nous donnons également beaucoup. Pourquoi aller développer ce qui existe déjà et qui est de très bonne qualité ? Linux n´est pas en conflit avec nos propres systèmes d´exploitation, il est même plutôt complémentaire. » **Marc Joly**, directeur du département Linux d´IBM France

« Ne pas faire soi-même ce que les autres font mieux » semble être la devise du constructeur informatique.

Pour une firme ayant des activités software uniquement, seuls les services et les plus produits (interface, addons, installation simplifiée, manuel d'utilisation, …) peuvent être source de revenus car la différenciation technologique est inexistante à cause de la liberté du code source.

RedHat distribue des solutions basées sur le noyau Linux et des services pour professionnels et particuliers. Or, linux est un logiciel sous licence GPL et celle-ci s'impose donc aux produits RedHat, qui ne peut ainsi implanter de produits propriétaires à ses distributions. La firme ne peut donc tirer de revenus que de produits annexes (I B).

Sur son segment d'activité principale (les entreprises), la stratégie du leader* est en phase d'évolution. Après s'être implantée à l'aide d'une politique de prix très agressive (coût d'installation réduit, services offerts pour essaie…) comme ses principaux concurrents, la firme envisage d'augmenter significativement les prix de son offre entreprise : le prix d'installation du logiciel reste certes peu élevé, mais le service d'assistance, de maintenance ou de mise à jour (compatibilité) se paie au tarif fort.

Microsoft qui voit d'un mauvais oeil la concurrence de linux, a déjà commencé aux Etats-Unis une campagne marketing comparant les coûts totaux (installation et maintenance) des produits pour entreprise. La firme s'apprête à faire de même en France avec une campagne publicitaire intitulée "Get the facts". En effet, si l'acquisition des logiciels est coûteuse dans le cas de Microsoft, les services et les coûts annexes sont moindres [voir tarifs des solutions en annexe]

RedHat voit sa stratégie menacée également par sa concurrence directe, les améliorations du logiciel faites par la firme sont librement appropriables par les challengers, qui poussent les prix vers le bas.

---

*chiffre d'affaires annuel : 126,1 millions de dollars, en hausse de 39%.(2004, EETIMES).

Nous voyons donc que les services peuvent rendre le libre viable, mais leurs coûts ne doivent pas freiner l'attrait du libre pour les clients et ne doivent pas inciter les outsiders ou challengers à une guerre des prix.

Il nous faut maintenant nous intéresser aux anticipations des firmes propriétaires (Sun, Apple…) qui ouvrent le code source de logiciel phare. Car à côté du développement de logiciels qui sont développés d'origine sous forme de logiciels libres (linux, apache…), de nouveaux OSS apparaissent à partir de la transformation de Code Source auparavant «privés » en logiciels libres. Nous pouvons citer notamment Apple qui a récemment ouvert (en partie) son system d'exploitation Mac OS X (projet Darwin).

Quelles motivations peuvent inciter les entreprises commerciales à adopter une stratégie Open Source, libérant ainsi, en partie, leurs secrets technologiques?
L'ouverture du code source conduit à une évolutivité plus importante du logiciel. Grâce à la contribution de la communauté du libre capable d'améliorer et d'assurer une compatibilité maximale le produit obtient des possibilités de développement plus importantes, ce qui assure une pérennité accrue du logiciel. Ainsi, une firme peut voir son produit amélioré (performance, simplicité d'usage…) par l'ouverture de son code source, sans pour autant investir massivement en R&D (les contributions du monde du libre compensant). Eventuellement, elle peut bénéficier d'une hausse des ventes d'un produit dont la qualité s'améliore ou dont l'évolutivité est garantie (certaines entreprises souhaitent avoir accès au code source afin de paramétrer au mieux le logiciel ; sécurité, confidentialité, réseau…).

Mais la transition du Closed à l'Open Source n'a pas que des avantages. La concurrence acquiert de façon très aisée les connaissances de la firme en programmation, cela pouvant combler un retard des challengers ou même leur octroyer des avantages (apprentissage de techniques de programmation qu'ils ne possèdent pas). De plus, la communauté du libre doit être incitée à contribuer au projet et ce sur le long terme. Car sans contribution, la R&D est obligatoire pour faire évoluer le logiciel, R&D bénéficiant au concurrent qui s'approprie vos secrets de fabrication.

La stratégie optimale de transition doit donc fournir des barrières à l'entrée (un plus face aux concurrents potentiels), et être compatible avec les attentes du monde du libre.

Ainsi Apple, en ouvrant uniquement le noyau de son Mac OS X et en gardant sous licence propriétaire l'interface et des composantes annexes (NeXT, Quick Time…) conserve les plus de son produit, limitant ainsi la concurrence (celle-ci doit investir pour programmer ses propres composantes protégées). (cf. : *Apple et l'open source : le projet Darwin* en annexe)

Le noyau étant quasiment identique pour les serveurs et les postes de travail, les contributions améliorant l'un sont donc bénéfiques à l'autre. Apple répondant au désir d'utilisateur/programmeur d'avoir accès au code source afin de personnaliser, de sécuriser…le système d'exploitation, s'assure par là même un certain niveau de contribution. Ces utilisateurs expérimentés améliorent l'OS serveur, et Apple vend des ordinateurs personnels avec un OS débogué. Cette stratégie répond au besoin des firme (plus de possibilité de programmation) et indirectement aux besoins des particuliers (simplicité et fiabilité).

Le projet Darwin semble donc être une stratégie efficace car assurant les critères de dissuasion à l'entrée (Apple mixant licence propriétaire et copyleft) et d'incitation à contribuer.

Les divers points énoncés nous laissent à penser qu'une stratégie d'entreprise alliant le libre peut être efficiente si elle a été correctement étudiée (choix licence, méthode d'appropriation de revenu, de protection…).

Avant de conclure, il nous apparaît qu'une question reste en suspens **: *L'open source est-il menacé par les brevets ?***

Si ce mémoire avait pour objectif de faire une revue de la littérature, nous ne pouvons omettre au moment de sa rédaction l'actualité du sujet tant sur un plan économique que juridique.

Nous avons déjà évoqué dans le texte la question de la brevetabilité du logiciel, comme une menace pour le libre. En mai 2004, la Commission et le Conseil des Ministres Européens examinent une loi sur la brevetabilité logicielle, sous la pression de multinationales et de juristes en brevets. Ce projet sous forme d'amendement à la directive "*sur la brevetabilité des inventions mises en oeuvre par ordinateur*" aurait pour effet de rendre illégaux des logiciels déjà développés et implanter, ces derniers utilisant des fonctionnalités, des modules, ou des routines voire des logiques de programmation….brevetables par les créateurs influent (Microsoft, IBM, Apple…) pionniers dans le développement.

Ainsi, le logiciel souvent inspiré de logiciel propriétaire (OpenOffice reprend certaines caractéristiques de Microsoft Office….) peut donc voir son évolution stoppée par une telle décision selon Mandrakesoft (fournisseur de distributions et de services Linux) :

" *Mandrakesoft tient à attirer l'attention de ses utilisateurs et de l'ensemble de l'industrie du logiciel sur l'adoption de la directive européenne sur les brevets logiciels.*
*En totale contradiction avec un récent vote du Parlement Européen mettant une limite claire à la brevetabilité des logiciels, la Présidence Irlandaise du Conseil des Ministres Européens prévoit d'introduire une brevetabilité illimitée. La décision actuelle doit être adoptée *sans débats* lors de la prochaine réunion des ministres les 17 et 18 mai prochain.*

Le nouveau texte, s'il est adopté en l'état, étendra les possibilités de brevetabilité à l'ensemble des composants logiciels y compris les programmes, les structures de données et les descriptions de processus. Cette décision nuira directement à la plupart des sociétés qui développent des logiciels ainsi qu'à l'ensemble des projets de Logiciel Libre qui n'ont pas les moyens de payer un droit de licence aux détenteurs des brevets. Cela se traduira par une appropriation du domaine public par des intérêts privés et une dérive où les industriels qui innovent, et créent des richesses et des emplois seront soumis à un racket de société ayant pour seule activité de déposer des portefeuilles de brevets logiciels sans créer aucun emploi ni innovation.

Notons que dans la liste des « sponsors » de la Présidence Irlandaise, on retrouve la société Microsoft, qui possède un portefeuille considérable de brevets, qu'elle pourrait utiliser en cas d'acceptation de cette proposition.

Mandrakesoft demande à ses utilisateurs de se mobiliser contre la menace grave et réelle que constitue une telle législation. Contactez les médias, vos élus locaux et votre gouvernement afin que ceux-ci votent contre cette proposition de brevetabilité sans limite et pousse les autorités européennes à revenir à la position démocratiquement adoptée par le Parlement Européen. " Source : PcInpact.com

Nous voyons au travers de ce qui précède que la viabilité du libre semble ne plus assuré dans le cas d'un tel projet de loi.

Ainsi le recours à stratégie « libre », dont nous avons essayé d'énoncer les avantages au fil des sections, se voit soumise à des aléas juridiques. (Foray, Zimmermann, 2002)

Conclusion :

Le logiciel libre grâce à « ses bonnes caractéristiques » (aspect libre et communautaire) bénéficie d'une notoriété croissante. Si ce bien peut être perçu à première vue comme un produits non marchand, son caractère économique, bien réel, fait de l'Open Source, une industrie extrêmement intéressante pour les chercheurs économistes. Elle permet de mettre en relation le juridique et l'économique (licences…), d'étudier la génération de revenus annexe tout en prenant en compte le rôle de l'incitation, de coordination et de coopération à un bien public. De plus, nous avons vu dans la seconde partie les applications de stratégie Open Source, nous montrant qu'une approche du libre peut être viable tout en étant contraignante et risquée pour les firmes y ayant recours.

Une stratégie utilisant le logiciel libre peut donc être une alternative pertinente face à une logique propriétaire, mais l'Open Source reste menacé par un régime de protection de la propriété intellectuelle en pleine évolution.

Bibliographie :

Firme et génération de revenus

Linus Dahlander, 2004 "Appropriating Returns from Open Innovation Processes: A Multiple Case Study of Small Firms in Open Source Software"

OSS et choix de licence

Josh Lerner, Jean Tirole, 2002 "The Scope of Open Source Licensing**"**

Bien public

David P. Myatt et Chris Wallace, 2002 « Equilibrium selection and public good provision: the development of open source software »

Texte additionnel:

Josh LERNER, Jean TIROLE, 2000 « The simple economics of Open Source »

James BESSEN, 2002 « Open Source Software: free provision of complex public goods »

Jacques CREMER, Alexandre GAUDEUL, 2003 « L'économie du logiciel libre »

Nicolas JULLIEN, Mélanie CLÉMENT-FONTAINE, Jean-Michel DALLE, 2002, « Projet RNTL : Nouveaux modèles économiques, nouvelle économie du logiciel ».

Eric S. RAYMOND, 1998, « La cathédrale et le bazar »

Joel WEST, 2002 "How Open is Open enough? Melding Proprietary and Open Source Platform Strategies"

Pio BAAKE et Thorsten WICHMANNY. « Open Source Software, Competition and Potential Entry »

Dominique FORAY, Jean-Benoît ZIMMERMANN, 2002 « L'économie du logiciel libre : Organisation coopérative et incitation à l'innovation »

Richard E. HAWKINS, 2002 « The economics of open source software for a competitive firm »

Liens Internet :

Labo de recherche sur le libre du MIT
http://opensource.mit.edu/

Hack Lab du Libre : site francophone de recherche
http://www.freescape.eu.org/biblio/rubrique.php3?id_rubrique=13

Première ressource francophone sur l'économie du Libre
http://www.logiciellibre.net/

Quand le brevet menace le logiciel :
http://swpat.ffii.org/papers/europarl0309/cons0401/index.fr.html
http://strasbourg.linuxfr.org

---

http://opensource.org/

http://www.indexel.net

http://www.linux.org/

http://www.redhat.com/

http://www.ibm.com/fr/

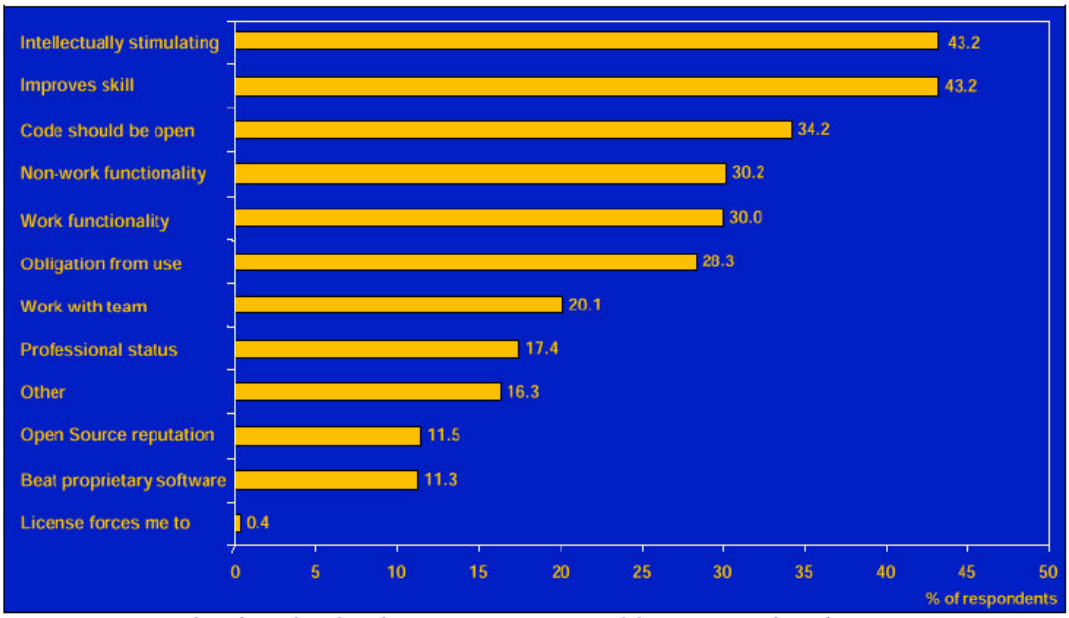
Annexes :

Résultat financier d'IBM

	2002	2001	2000
Revenues			
Global Services	36360	34956	33152
Hardware	27456	30593	34470
Software	13074	12939	12598
Global Financing	3232	3426	3465
Enterprise Investments / Other	1064	1153	1404
Total	**81186**	**83067**	**85089**
Dépenses			
Global Services	26812	25355	24309
Hardware	20020	21231	24207
Software	2043	2265	2283
Global Financing	1416	1693	1965
Enterprise Investments / Other	611	634	747
Total	**50902**	**51178**	**747**
Marge brute			
Global Services	9548	9601	8843
Hardware	7436	9362	10263
Software	11031	10674	10315
Global Financing	1816	1733	1500
Enterprise Investments / Other	453	519	657
Total	**30284**	**31889**	**31578**

Source : IBM

Motivation des membres de la communauté au développement du libre :



Motivation des hackers, Boston Consulting Group, janvier 2002

Source : « La dynamique *open source* » Robert Viseur, 2002

Le libre en chiffre

Part de marché des distributions linux (pour le Royaume-Uni)

- Mandrake (46,02%)
- RedHat (21,33%)
- SuSE (18,67%)
- Debian (5,33%)

Source : Linux Magazine UK, 2001

Les systèmes d'exploitations des serveurs dans le monde en 2002:

- Windows (55,2%)
- Linux (23,1%)
- Unix (11,0%)
- Netware (9,9%)
- Autres (7,3%)

Source : IDC, 2003

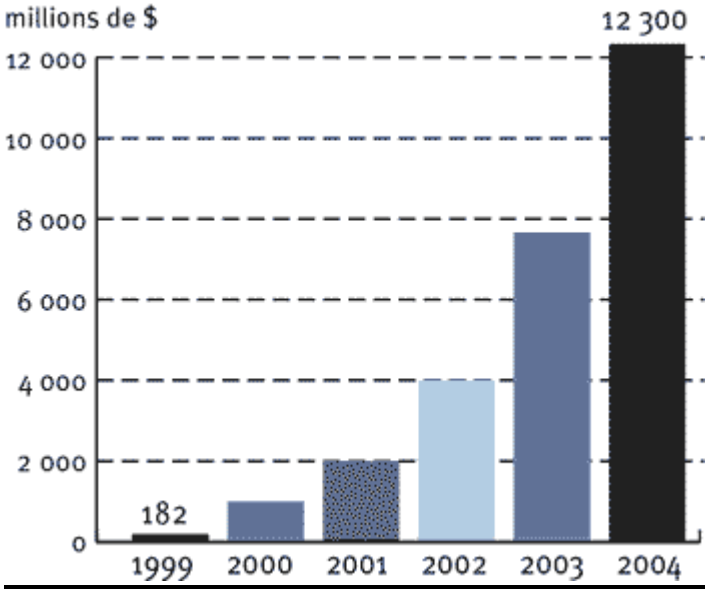
Les données suivantes peuvent être erronées, les divergences dues au mode de diffusion des logiciels libres :

- 40% des décisionnaires utilisent Linux. (Août 2001, IDC 800 interrogés)
- 8,6% (Gartner, 2001) 24% (IDC, 2001) des serveurs fonctionnent sous Linux
- 60% des sites Internet administrés sous Apache
- 4% des postes de travail fonctionnent sous une distribution Linux

Mandrake génération des revenus :

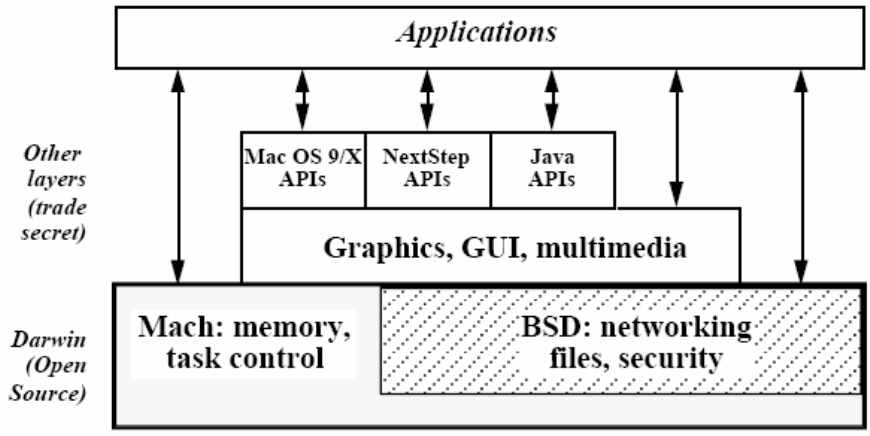
- *utilitaires* (97 % du CA) : facilitant l'installation, la configuration et l'administration du noyau du système d'exploitation Linux.
- *services* (3 %) : support Web et téléphonique à accès limité.

Linux une évolution intéressante :



Revenu des distributions linux (source : google.com)

Apple et l'open source : le projet Darwin



Source: Apple Computer (2000)

Figure 1: Architectural layers for Apple's Mac OS X platform

Mac OS X : un système d'exploitation propriétaire « ouvert »

Prix d'une distribution Linux

Le système en lui-même est gratuit. Mais les éditeurs font bien évidemment payer le support qu'ils proposent. Les tarifs s'appliquent par serveur ou par poste, sans notion de nombre d'utilisateurs ni de processeurs. Il est donc possible de télécharger une Red Hat sans frais. Mais pour la version Advanced Server destinée aux entreprises, l'éditeur propose trois niveaux et donc trois tarifs de support. Le plus basique revient à environ **900 dollars par an** et par machine, le plus complet à **2 800 dollars.**

Comparatif des systèmes d'exploitation

Tarif indicatif pour des versions boites (monoposte, ni MAJ, ni OEM) *mai 2004, LDLC.com*

APPLE	
Apple MAC OS X - Version 10.2	149€

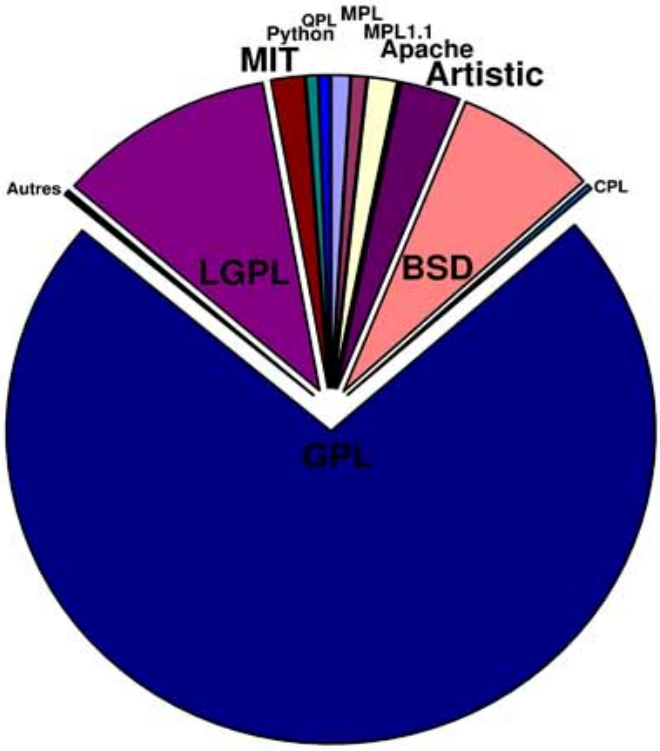
MANDRAKE	
Mandrake Discovery 9.2 (OS & suite logiciel)	39€*
Mandrake Linux PowerPack 9.2 (OS & suite logiciel)	79 €*
Mandrake Linux Pro suite 9.2 (OS & suite logiciel)	199€*

MICROSOFT	
Microsoft Windows XP Edition Familiale (OS)	269 €
Microsoft Windows XP Edition Professionnelle (OS)	399€
Microsoft Windows 2003 Server Entreprise (OS)	890€
Microsoft Office 2003 standard (suite logiciel)	579€

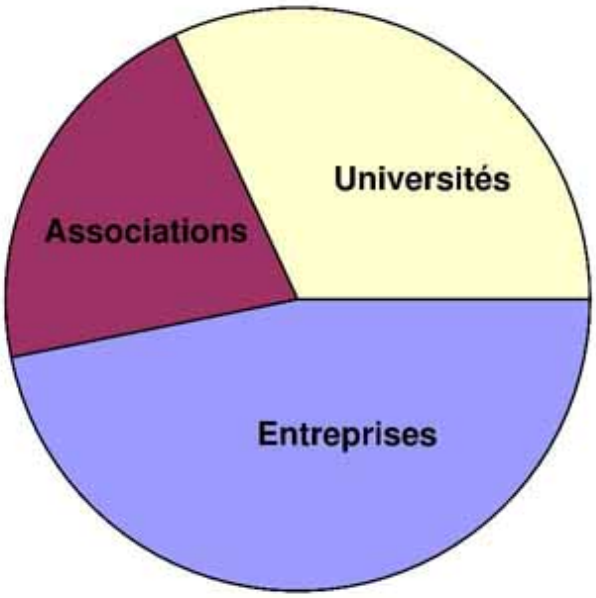
RED HAT
Red Hat Linux 9.0 Personal (OS & suite logiciel) 65,00 €*
Red Hat Linux 9.0 Professional (OS & suite logiciel) 299,00 €*

* support et assistance inclus (durée variant selon les versions)

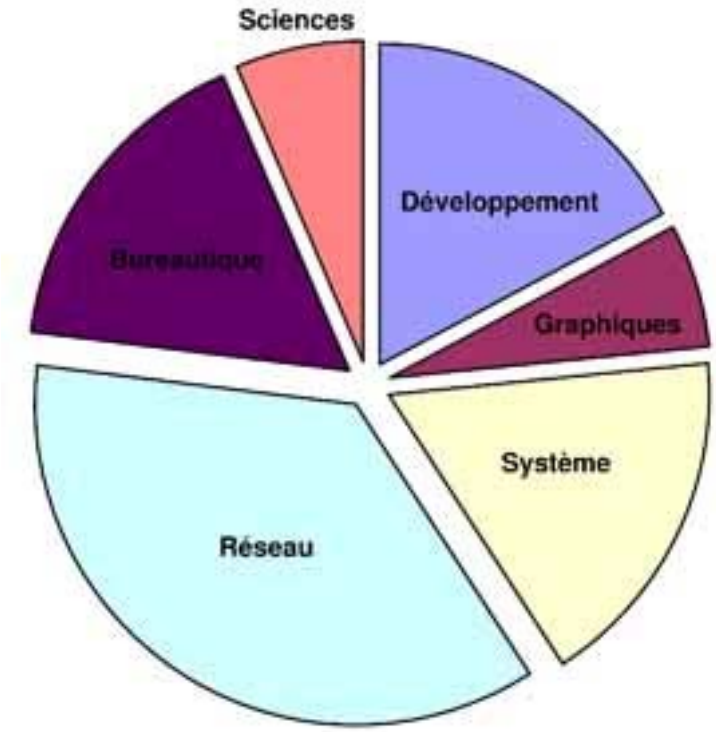
Les licences



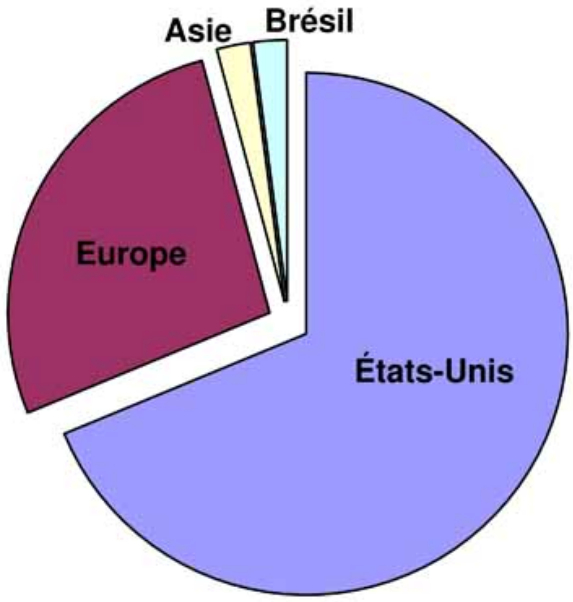
Répartition des licences Open Source par type



Origine des contributions Open Source



Répartition de programme Open Source par thème



Origine géographique des contributions Open Source

Source :

Nicolas JULLIEN, Mélanie CLÉMENT-FONTAINE, Jean-Michel DALLE, 2002, « Projet RNTL : Nouveaux modèles économiques, nouvelle économie du logiciel ».